» User Guide «

AM5020
uEFI BIOS

Doc. ID: 1045-0247, Rev. 1.0
April 21, 2011

Revision History

Publication Title:	AM5020 uEFI BIOS User Guide	
Doc. ID:	1045-0247	
Rev.	Brief Description of Changes	Date of Issue
1.0	Initial issue based on the uEFI BIOS version R13	21-Apr-2011

Imprint

Kontron Modular Computers GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Modular Computers GmbH

Sudetenstraße 7

D - 87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

+49 (0) 800-SALESKONTRON

sales@kontron.com

For further information about other Kontron products, please visit our Internet web site: www.kontron.com.

Disclaimer

Copyright © 2011 Kontron AG. All rights reserved. All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

Table of Contents

Revision History ..... *ii*
Imprint ..... *ii*
Disclaimer ..... *ii*
Table of Contents ..... *iii*

1. Starting uEFI BIOS Setup 3
1.1 Main Setup Menu 4
1.2 Navigation 5

2. Main Setup 9
2.1 BIOS Information 9
2.2 UnCore Information 9
2.3 Trusted Computing 10
2.3.1 TPM Configuration 10
2.3.1.1 TPM Support 10
2.4 USB Configuration 11
2.4.1 USB Configuration 11
2.4.2 Legacy USB Support 11
2.4.3 EHCI Hand-Off 12
2.4.4 Device Reset Timeout 12
2.4.5 Mass Storage Devices 12
2.5 Serial Port Console Redirection 13
2.5.1 COM0 13
2.5.1.1 Console Redirection 13
2.5.1.2 Console Redirection Settings 13
2.5.2 COM4 14
2.5.2.1 Console Redirection 14
2.5.2.2 Console Redirection Settings 14
2.5.3 Serial Port for Out-of-Band Management/Windows EMS 14
2.5.3.1 Console Redirection 14
2.5.3.2 Out-of-Band Mgmt Port 14
2.5.3.3 Data Bits 14

2.5.3.4 Parity 15
2.5.3.5 Stop Bits 15
2.5.3.6 Terminal Type 15
2.5.4 Console Redirection Settings 16
2.5.4.1 Terminal Type 16
2.5.4.2 Bits per second 16
2.5.4.3 Data Bits 17
2.5.4.4 Parity 17
2.5.4.5 Stop Bits 17
2.5.4.6 Flow Control 17
2.5.4.7 Resolution 100x31 17
2.5.4.8 Legacy OS Redirection 18
2.6 System Language 18
2.7 System Date 18
2.8 System Time 18
2.9 Access Level 18

3. Boot Setup 21

3.1 Boot Configuration 21
3.1.1 Quiet Boot 21
3.1.2 Fast Boot 22
3.1.3 uEFI Boot 22
3.1.4 Setup Prompt Timeout 22
3.1.5 Bootup NumLock State 22
3.1.6 CSM16 Module Version 22
3.1.7 GateA20 Active 23
3.1.8 Option ROM Messages 23
3.1.9 Interrupt 19 Capture 23
3.2 Boot Option Priorities 24
3.2.1 Boot Option #1..2 24
3.2.2 Hard Drive/Network Device/CD/DVD ROM Drive/Floppy Drive/etc.. 24
3.2.3 Add New Boot Option 24
3.2.4 Delete Boot Option 24

4. Security Setup 27
4.1 Administrator Password 28
4.2 User Password 28
4.3 Remember the Password 28

5. Save & Exit 31
5.1 Save Changes and Exit 31
5.2 Discard Changes and Exit 31
5.3 Save Changes and Reset 31
5.4 Discard Changes and Reset 32
5.5 Save Changes (Save Options) 32
5.6 Discard Changes (Save Options) 32
5.7 Restore Defaults (Save Options) 32
5.8 Save as User Defaults (Save Options) 32
5.9 Restore User Defaults (Save Options) 32
5.10 Boot Override 32

6. The uEFI Shell 35
6.1 Introduction, Basic Operation 35
6.1.1 Shell Startup 35
6.2 Kontron-Specific uEFI Shell Commands 36
6.2.1 kboardconfig uEFI Shell Command 37
6.2.2 kboardinfo uEFI Shell Command 39
6.2.3 kboot uEFI Shell Command 41
6.2.4 kbootnsh uEFI Shell Command 43
6.2.5 kclearnvram uEFI Shell Command 44
6.2.6 kclsp uEFI Shell Command 44
6.2.7 kflash uEFI Shell Command 45
6.2.8 kipmi uEFI Shell Command 46
6.2.9 kmkramdisk uEFI Shell Command 49
6.2.10 kpassword uEFI Shell Command 50
6.2.11 kpci uEFI Shell Command 51
6.2.12 kwdt uEFI Shell Command 52

6.3 uEFI Shell Scripting 54
6.3.1 Startup Scripting 54
6.3.2 Create a Startup Script 54
6.3.3 Examples of Startup Scripts 54
6.3.3.1 Automatic Booting from USB Flash Drive 54
6.3.3.2 Switch On Clock Spreading Prior to Booting from Harddrive 54
6.3.3.3 Execute Shell Script on Other Harddrive 54
6.3.3.4 Enable Watchdog and Control PXE Boot 55
6.3.3.5 Handling the Startup Script in the Flash Bank 56

7. Updating the uEFI BIOS 59
7.1 BIOS Redundancy Strategy 59
7.2 Updating Strategy 59
7.3 uEFI BIOS Recovery 59
7.4 Determining the Active Flash 59
7.5 Manual Flash Selection 60
7.6 Flash Selection by DIP Switch 60

Chapter 1

Starting uEFI BIOS Setup

This page has been intentionally left blank.

1. Starting uEFI BIOS Setup

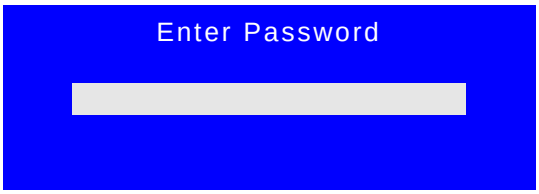
The AM5020 is provided with a Kontron-customized, pre-installed and configured version of Aptio® (referred to as uEFI BIOS in this manual), AMI's next generation BIOS firmware based on the Unified Extensible Firmware Interface (uEFI) specification and the Intel® Platform Innovation Framework for EFI. This uEFI BIOS provides a variety of new and enhanced functions specifically tailored to the hardware features of the AM5020. This user guide reflects the uEFI BIOS version R13.

To take advantage of these functions, the uEFI BIOS comes with a Setup program which provides quick and easy access to the individual function settings for control or modification of the uEFI BIOS configuration.

The Setup program allows the accessing of various menus which provide functions or access to sub-menus with more specific functions of their own. The individual menus and the configurable functions are described in this guide.

To start the uEFI BIOS Setup program, follow the steps below:

1. Power on the board.
2. Wait until the first characters appear on the screen (POST messages or splash screen).
3. Press the <F2> key.
4. If the uEFI BIOS is password-protected, a window such as the one below will appear:



   Enter either the User password or the Administrator password (refer to Chapter 4, Security Setup, for further information), press <RETURN>, and proceed with step 2.

5. A Setup menu with the following token attributes will appear.
   The currently active menu and the currently active uEFI BIOS Setup item are highlighted in white.

Note: Both the uEFI BIOS Setup and the uEFI Shell are accessible via the serial port.

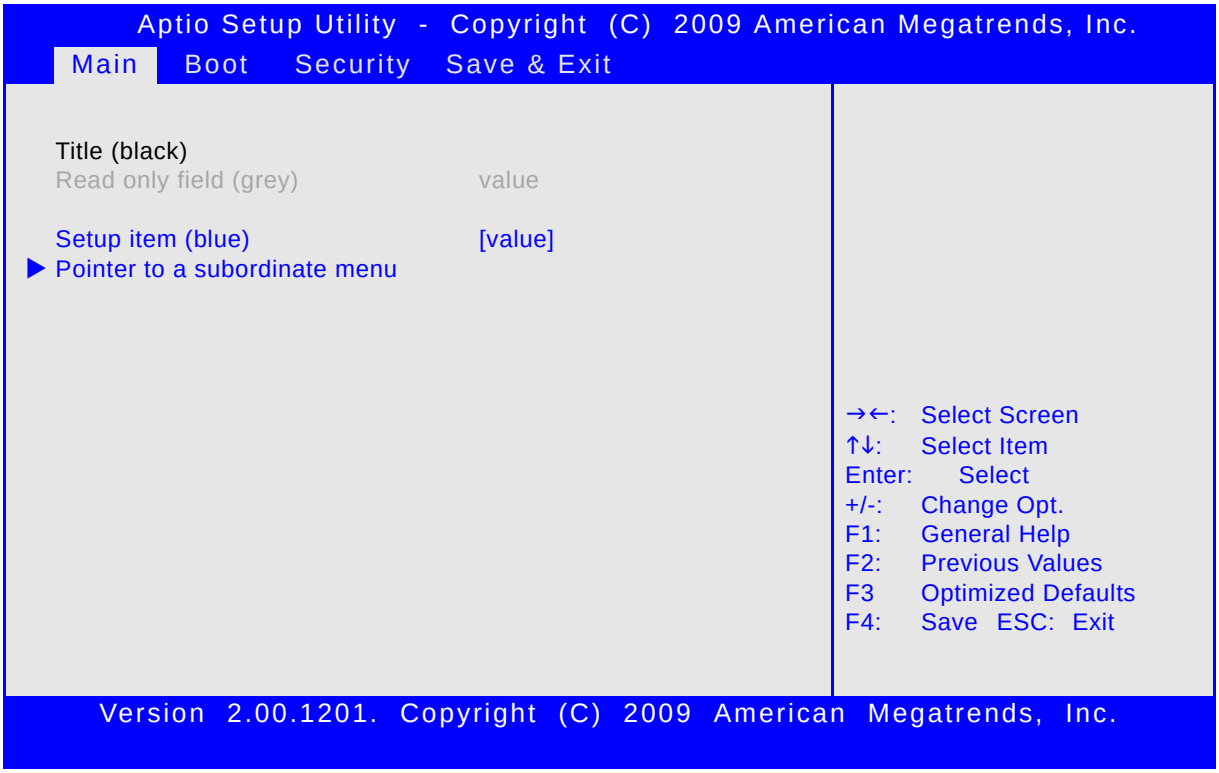
1.1 Main Setup Menu

The Main setup menu is the first screen that appears after starting the Setup program.

At the top of this screen and all of the other major screens, there is a setup menu selection bar, which permits access to all of the other major setup menus. These menus are selected via the left-right arrow keys.

All setup menu screens have two main frames. The left frame displays all the functions that can be configured. They are displayed in blue. Functions displayed in gray provide information about the status or the operational configuration.

The right frame displays the key legend. Above the key legend there is an area reserved for a text message. When a function is selected in the left frame, it is displayed in white. Often a text message will accompany it.



1.2 Navigation

The AM5020 uEFI BIOS setup program uses a hot key-based navigation system. A hot key legend is located in the right frame on most setup screens.The following table provides information concerning the usage of these hot keys.

HOT KEY	DESCRIPTION
<F1>	The <F1> key is used to invoke the General Help window.
<F2>	The <F2> key is used to restore the previous values.
<F3>	The <F3> key is used to load the defaults.
<F4>	The <F4> key is used to save the current settings and exit the uEFI BIOS Setup.
→ ← Left/Right	The *Left and Right* <Arrow> keys are used to select a major Setup screen. For example: Main Screen, Advanced Screen, Chipset Screen, etc.
↑ ↓ Up/Down	The *Up and Down* <Arrow> keys are used to select a Setup function or a sub-screen.
+ - Plus/Minus	The *Plus and Minus* <Arrow> keys are used to change the field value of a particular Setup function, for example, system date and time.
<ESC>	The <ESC> key is used to exit a menu or the uEFI BIOS Setup. Pressing the <ESC> key in a sub-menu causes the next higher menu level to be displayed. When the <ESC> key is pressed in a major Setup menu, the uEFI BIOS Setup is terminated without saving any changes made.
<Enter>	The <Enter> key is used to execute a command or select a menu.

This page has been intentionally left blank.

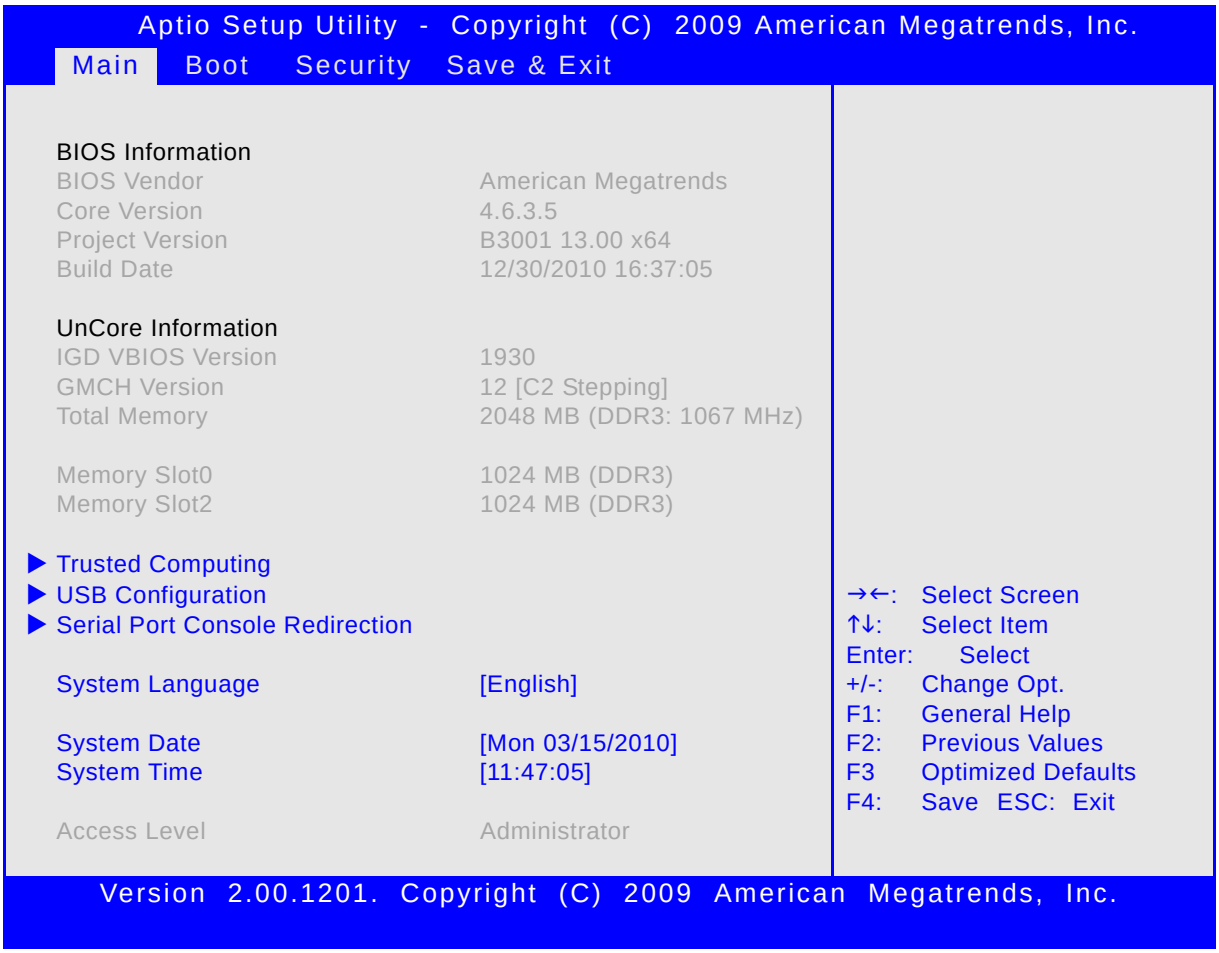
Chapter 2

Main Setup

This page has been intentionally left blank.

2. Main Setup

Upon entering the uEFI BIOS Setup program, the Main setup screen is displayed. This screen lists the main setup sub-screens and provides very basic system information as well as functions for setting the system time and date. In addition, the remaining major setup menus can be accessed from this screen. This screen can also be selected from any other major setup screen by using the Main tab.


Aptio Setup Utility - Copyright (C) 2009 American Megatrends, Inc.
Main Boot Security Save & Exit
BIOS Information
BIOS Vendor American Megatrends
Core Version 4.6.3.5
Project Version B3001 13.00 x64
Build Date 12/30/2010 16:37:05
UnCore Information
IGD VBIOS Version 1930
GMCH Version 12 [C2 Stepping]
Total Memory 2048 MB (DDR3: 1067 MHz)
Memory Slot0 1024 MB (DDR3)
Memory Slot2 1024 MB (DDR3)
Trusted Computing
USB Configuration
Serial Port Console Redirection
System Language [English]
System Date [Mon 03/15/2010]
System Time [11:47:05]
Access Level Administrator
→←: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3 Optimized Defaults
F4: Save ESC: Exit
Version 2.00.1201. Copyright (C) 2009 American Megatrends, Inc.

2.1 BIOS Information

This function provides display-only information concerning the uEFI BIOS.

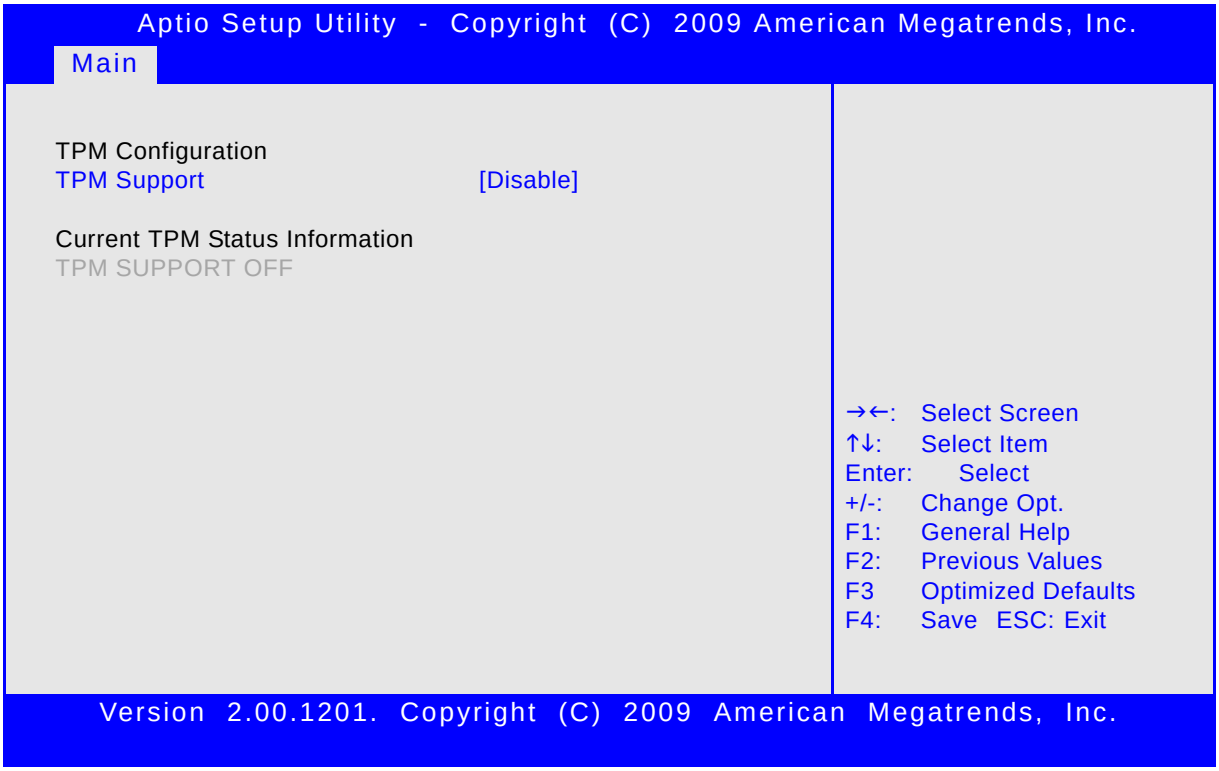
Information about the running uEFI BIOS version is reflected in the display-only function Project Version (parameter “13.00” indicates Rev. 13).

2.2 UnCore Information

This function provides display-only information concerning the NorthBridge (GMCH die of the Intel® Core™ i7 processor) features and the system memory.

2.3 Trusted Computing

This screen provides functions for specifying the TPM configuration settings and TPM displaying status information.



2.3.1 TPM Configuration

2.3.1.1 TPM Support

This function is used to provide the Trusted Platform Module (TPM) functionality to the OS.

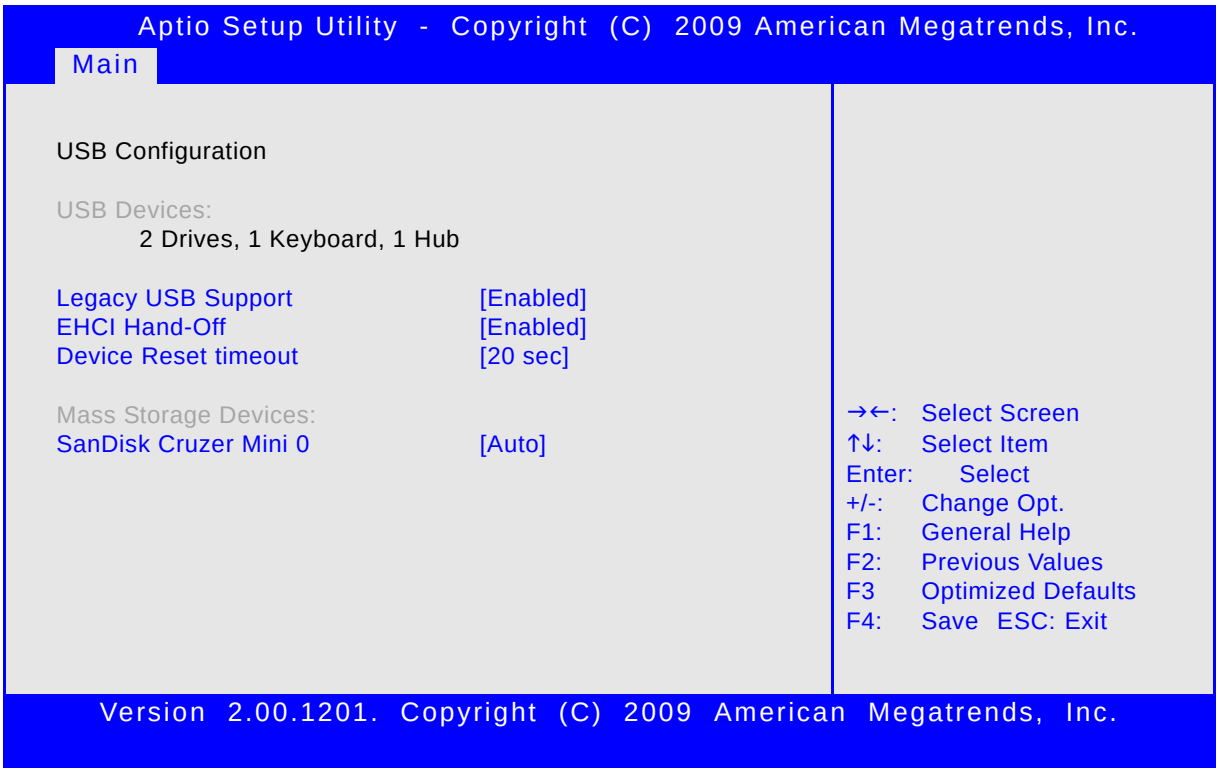
Note: Trusted Platform Module support is available on request.

SETTING	DESCRIPTION
Disable	Use this setting to disable TPM support. If this setting is used, TPM is not present for the OS, regardless whether the function TPM State is enabled or not.
Enable	Use this setting to enable TPM support.

Default setting: Disable

2.4 USB Configuration

This screen provides information about support for USB devices as well as functions for specifying the USB configuration settings.



2.4.1 USB Configuration

This is a display-only function providing general information about the USB devices detected.

2.4.2 Legacy USB Support

This function is required for booting from USB devices and for operating systems which do not support USB themselves (mainly DOS and some BootLoaders).

SETTING	DESCRIPTION
Disabled	Use this setting to disable legacy USB support.
Enabled	Use this setting to enable legacy USB support.
Auto	Use this setting to enable legacy USB support if there are USB devices present.

Default setting: Enabled

2.4.3 EHCI Hand-Off

This function is used to enable a workaround for operating systems without EHCI Hand-Off support. The EHCI ownership change should be claimed by the EHCI driver.

Note: It is recommended to leave this function at the default setting.
For operating systems without USB2.0 support this function must be left at the default setting.

SETTING	DESCRIPTION
Disabled	Use this setting to disable EHCI Hand-Off support.
Enabled	Use this setting to enable EHCI Hand-Off support.

Default setting: Enabled

2.4.4 Device Reset Timeout

This setting selects the timeout in seconds that the USB core will wait for a USB storage device to become ready after start unit command.

SETTING	DESCRIPTION
10 sec 20 sec 30 sec 40 sec	Use one of these settings to specify how long the USB core will wait for a USB mass storage device to become ready after the start unit command.

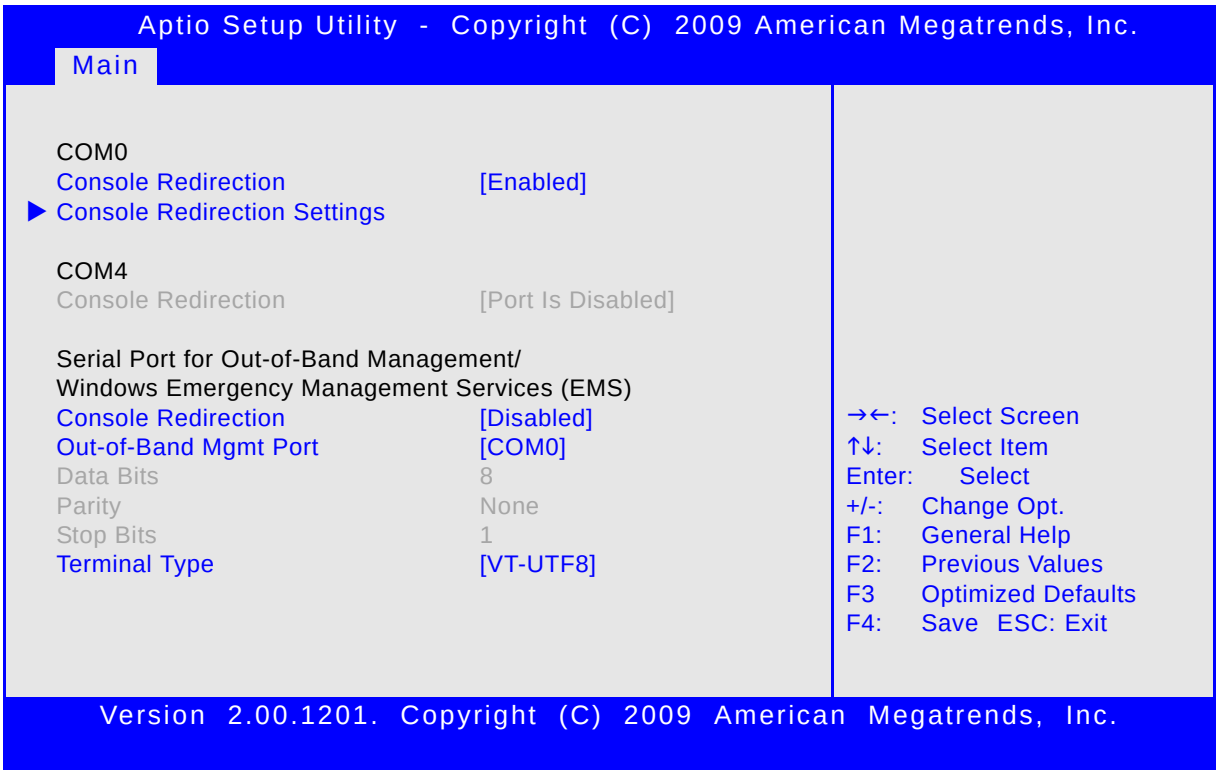
Default setting: 20 sec

2.4.5 Mass Storage Devices

This function shows a list of connected USB mass storage devices and allows the user to select how the respective device is to be treated.

2.5 Serial Port Console Redirection

This screen provides information about functions for specifying the Serial Port Console Redirection configuration settings. Console redirection can be used to remotely operate system settings and the EFI console.



2.5.1 COM0

The COM0 port (serial port 0) in the uEFI BIOS corresponds to the COMA serial port on the front panel of the AM5020.

2.5.1.1 Console Redirection

SETTING	DESCRIPTION
Disabled	Use this setting to disable console redirection for the serial port 0.
Enabled	Use this setting to enable console redirection for the serial port 0.

Default setting: Enabled

2.5.1.2 Console Redirection Settings

For information about this function, refer to Chapter 2.5.4 in this manual.

2.5.2 COM4

COM4 is available only if the MicroTCA system provides a serial port via PCI Express.

2.5.2.1 Console Redirection

SETTING	DESCRIPTION
Disabled	Use this setting to disable console redirection for a PCIe serial port.
Enabled	Use this setting to enable console redirection for a PCIe serial port.

Default setting: Enabled

2.5.2.2 Console Redirection Settings

For information about this function, refer to Chapter 2.5.4 in this manual.

2.5.3 Serial Port for Out-of-Band Management/Windows Emergency Management Services (EMS)

The following functions control the presence and content of the ACPI serial port redirection table (SPCR). This table is mainly used by the Windows server variants to provide Windows Emergency Management Services (EMS). This functionality is totally independent from serial redirection of other console output.

2.5.3.1 Console Redirection

SETTING	DESCRIPTION
Disabled	Use this setting to prevent the system from adding the SPCR table to the ACPI tables.
Enabled	Use this setting to add the SPCR table to the ACPI tables. The OS can further use the information provided for serial redirection services.

Default setting: Disabled

2.5.3.2 Out-of-Band Mgmt Port

This function is used to select the serial port intended for use with Out-of-Band Management. This functionality is independent from serial redirection of other console output.

SETTING	DESCRIPTION
COM0	Use this setting to specify that the serial port 0 is to be used with Out-of-Band Management.
COM4	Use this setting to specify that a PCIe serial port is to be used with Out-of-Band Management.

Default setting: COM0

2.5.3.3 Data Bits

This is a display-only function providing information about the frame width for the Out-of-Band Management.

2.5.3.4 Parity

This is a display-only function providing information about the parity for Out-of-Band Management.

2.5.3.5 Stop Bits

This is a display-only function providing information about the number of stop bits for Out-of-Band Management.

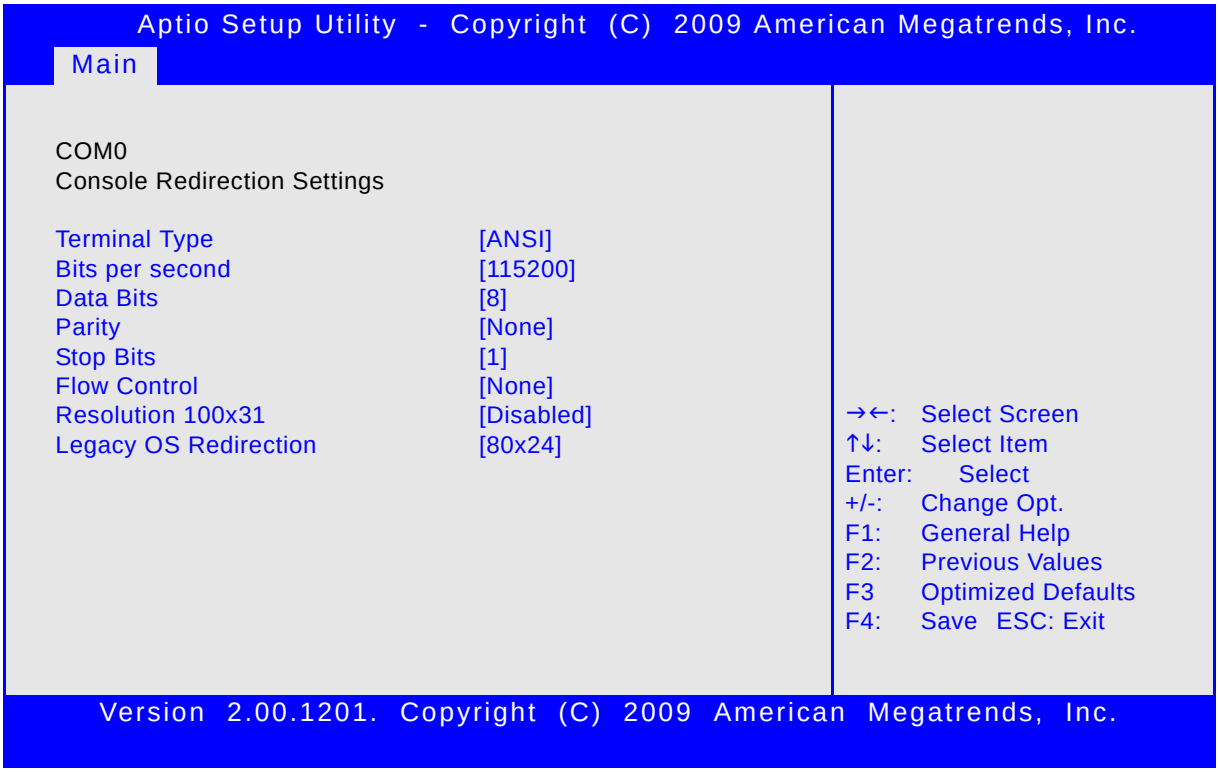
2.5.3.6 Terminal Type

SETTING	DESCRIPTION
VT100	Use one of these settings to select the terminal type for out-of-band management.
VT100+	
VT-UTF8	
ANSI	

Default setting: VT-UTF8

2.5.4 Console Redirection Settings

This screen provides information about functions for specifying the Console Redirection configuration settings for the serial port 0 and a PCIe serial port. Each serial port can be independently configured.



2.5.4.1 Terminal Type

SETTING	DESCRIPTION
VT100	Use one of these settings to select the terminal type to be emulated.
VT100+	
VT-UTF8	
ANSI	

Default setting: ANSI

2.5.4.2 Bits per second

SETTING	DESCRIPTION
9600	Use one of these settings to select the baud rate of the serial port.
19200	
57600	
115200	

Default setting: 115200

2.5.4.3 Data Bits

SETTING	DESCRIPTION
7	Use one of these settings to specify the number of data bits per frame.
8	

Default setting: 8

2.5.4.4 Parity

SETTING	DESCRIPTION
None	Use one of these settings to select the parity for the serial port.
Even	
Odd	
Mark	
Space	

Default setting: None

2.5.4.5 Stop Bits

SETTING	DESCRIPTION
1	Use one of these settings to specify the number of stop bits for the serial port.
2	

Default setting: 1

2.5.4.6 Flow Control

SETTING	DESCRIPTION
None	Use one of these settings to specify the type of flow control to be used for this serial port.
Hardware RTS/CTS	
Software Xon/Xoff	

Default setting: None

2.5.4.7 Resolution 100x31

SETTING	DESCRIPTION
Disabled	Use this setting the disable extended terminal resolution.
Enabled	Use this setting the enable extended terminal resolution.

Default setting: Disabled

2.5.4.8 Legacy OS Redirection

SETTING	DESCRIPTION
80x24	Use one of these settings to select the number of rows and columns for legacy OS redirection.
80x25	

Default setting: 80x24

2.6 System Language

SETTING	DESCRIPTION
English	Use this function to select the system language. Currently, only English is supported.

2.7 System Date

SETTING	DESCRIPTION
<WD MM/DD/YYYY>	Use this function to change the system date. Select System Date using the Up and Down <Arrow> keys. Enter the new values through the keyboard or press +/- to increment/decrement values. Use “Tab” to switch between date elements.

2.8 System Time

SETTING	DESCRIPTION
<HH:MM:SS>	Use this function to change the system time. Select System Time using the Up and Down <Arrow> keys. Enter the new values through the keyboard or press +/- to increment/decrement values. Use “Tab” to switch between time elements.

Note: The time is in 24-hour format. For example, 5:30 A.M. appears as 05:30:00, and 5:30 P.M. as 17:30:00.

2.9 Access Level

This function provides display-only information concerning the uEFI BIOS Setup accessibility for the current Setup session. Depending on the type of password protection used, one of the following settings is displayed:

SETTING	DESCRIPTION
Administrator	This setting indicates that read/write access to all setup options is available.
User	This setting indicates that only a limited subset of all setup options is modifiable.

Note: If no password is set, the access setup is Administrator.

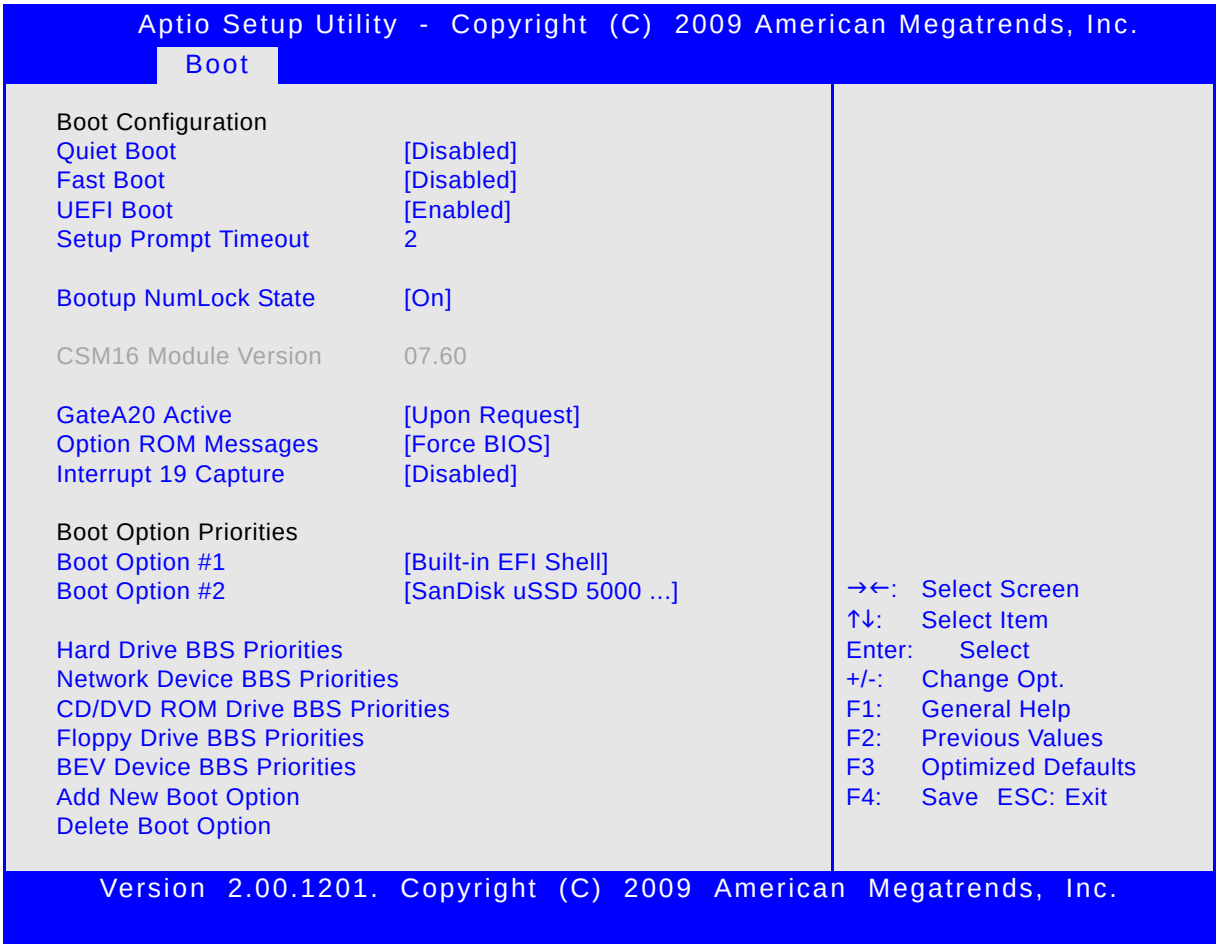
Chapter 3

Boot Setup

This page has been intentionally left blank.

3. Boot Setup

Select the Boot tab to enter the Boot Setup screen. This screen lists the sub-screens for boot configuration and boot device priority.


Aptio Setup Utility - Copyright (C) 2009 American Megatrends, Inc.
Boot
Boot Configuration
Quiet Boot [Disabled]
Fast Boot [Disabled]
UEFI Boot [Enabled]
Setup Prompt Timeout 2
Bootup NumLock State [On]
CSM16 Module Version 07.60
GateA20 Active [Upon Request]
Option ROM Messages [Force BIOS]
Interrupt 19 Capture [Disabled]
Boot Option Priorities
Boot Option #1 [Built-in EFI Shell]
Boot Option #2 [SanDisk uSSD 5000 ...]
Hard Drive BBS Priorities
Network Device BBS Priorities
CD/DVD ROM Drive BBS Priorities
Floppy Drive BBS Priorities
BEV Device BBS Priorities
Add New Boot Option
Delete Boot Option
→←: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3 Optimized Defaults
F4: Save ESC: Exit
Version 2.00.1201. Copyright (C) 2009 American Megatrends, Inc.

3.1 Boot Configuration

3.1.1 Quiet Boot

This function is used to display either POST output messages or a splash screen during boot-up.

SETTING	DESCRIPTION
Disabled	Use this setting to display POST output messages during boot-up.
Enabled	Use this setting to display a splash screen during boot-up.

Default setting: Disabled

3.1.2 Fast Boot

This function is used to enable or disable boot with initialization of a minimal set of devices required to launch active boot option..

SETTING	DESCRIPTION
Disabled	Use this setting to disable fast boot.
Enabled	Use this setting to enable fast boot.

Default setting: Disabled

3.1.3 uEFI Boot

This function is used to enable or disable uEFI boot from disks.

SETTING	DESCRIPTION
Disabled	Use this setting to prevent the system from booting native uEFI-aware operating systems from disks.
Enabled	Use this setting to enable booting of native uEFI-aware operating systems from disks, if present, and in boot order.

Default setting: Enabled

3.1.4 Setup Prompt Timeout

This integer function is used to set an additional time the POST should wait for the operator to press the key to enter setup. The time is entered in seconds.

SETTING	DESCRIPTION
1 ⋮ 65535	Use one of these settings to specify the setup prompt timeout.

Default setting: 2

3.1.5 Bootup NumLock State

This function is used to set the state of the keyboard’s numlock function after POST.

SETTING	DESCRIPTION
On	Use this setting to switch on the keyboard’s numlock function after POST.
Off	Use this setting to switch off the keyboard’s numlock function after POST.

Default setting: On

3.1.6 CSM16 Module Version

This function provides display-only information concerning the CSM Module and is intended for internal use only.

3.1.7 GateA20 Active

This function is used to enable or disable GateA20.

SETTING	DESCRIPTION
Upon Request	Use this setting to disable GateA20 in the uEFI BIOS.
Always	Use this setting to prevent the system from disabling GateA20.

Default setting: Upon Request

3.1.8 Option ROM Messages

This function is used to control the messages of the loaded PCI option ROMs.

SETTING	DESCRIPTION
Force BIOS	Use this setting to force to a BIOS-compatible output. This will show the option ROM messages.
Keep Current	Use this setting to keep the current video mode. This will suppress option ROM messages. Option ROMs requiring interactive inputs may not work properly in this mode.

Default setting: Force BIOS

3.1.9 Interrupt 19 Capture

This function is used to specify if legacy PCI option ROMs are allowed to capture software interrupt 19h.

SETTING	DESCRIPTION
Disabled	Use this setting to prevent legacy PCI option ROMs from capturing software interrupt 19h.
Enabled	Use this setting to allow legacy PCI option ROMs to capture software interrupt 19h.

Default setting: Disabled

3.2 Boot Option Priorities

3.2.1 Boot Option #1..2

These functions are used to form the boot order and are dynamically generated. They represent either a legacy BBS (BIOS Boot Specification) class of devices or a native EFI boot entry. Press Return on each option to select the BBS class / EFI boot entry desired.

3.2.2 Hard Drive/Network Device/CD/DVD ROM Drive/Floppy Drive/BEV Device BBS Priorities

These functions lead to sub-menus that allow configuring the boot order for a specific device class. These options are visible only if at least one device for this class is present. These functions are dynamically generated.

The PXE boot devices (network adapters) are controlled via the **kboardconfig** uEFI Shell command. The PXE boot devices are listed as boot devices after being enabled by the uEFI Shell.

3.2.3 Add New Boot Option

This function is used to create a native uEFI boot option and is visible only if at least one appropriate native boot device is present. Please refer to the documentation for the respective native uEFI-aware operating system for further information about creating a boot option.

3.2.4 Delete Boot Option

This function is used to delete a native uEFI boot option. Please refer to the documentation for the respective native uEFI-aware operating system for further information about deleting a boot option.

Note: Do not delete the “Built-in EFI Shell” boot option as this would remove the uEFI Shell from the boot order. In case the uEFI Shell got removed, use “Save & Exit” / “Boot Override” / “Built-in EFI Shell” to recover.

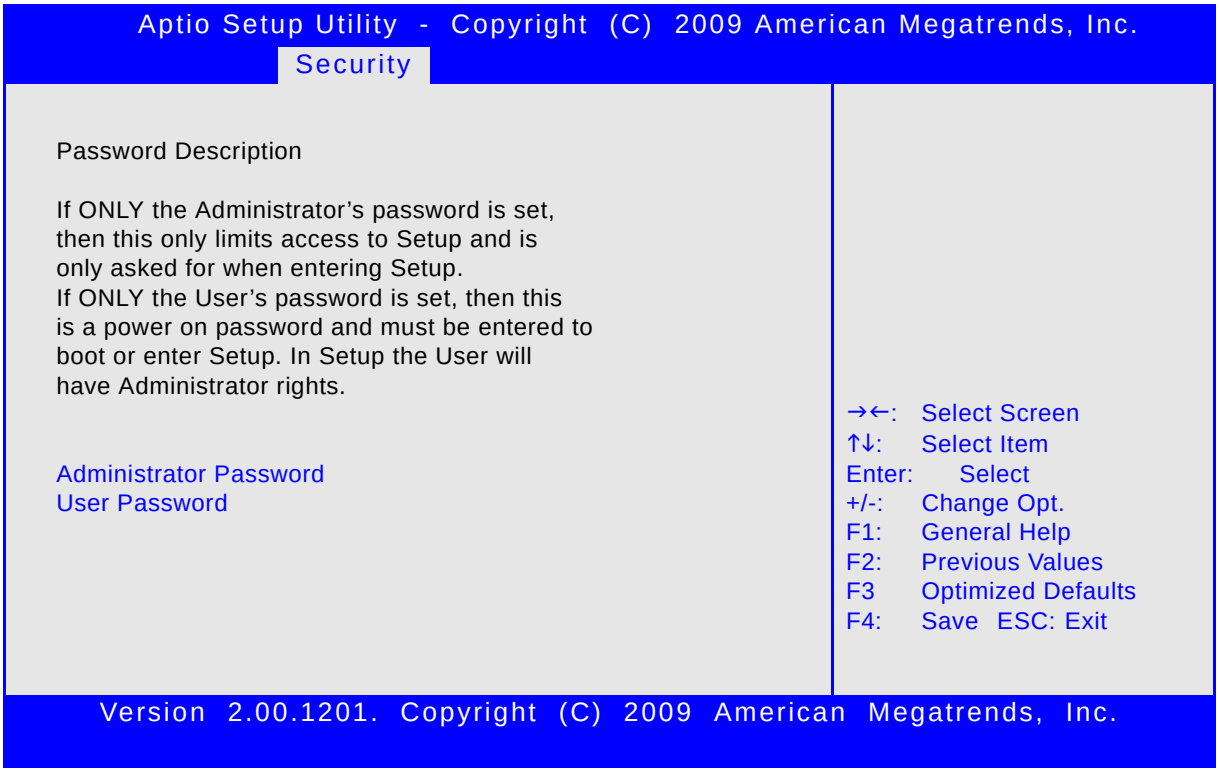
Chapter 4

Security Setup

This page has been intentionally left blank.

4. Security Setup

Select the Security tab to enter the Security Setup screen. This screen provides information about the passwords and functions for specifying the security settings.


Aptio Setup Utility - Copyright (C) 2009 American Megatrends, Inc.
Security
Password Description
If ONLY the Administrator's password is set, then this only limits access to Setup and is only asked for when entering Setup.
If ONLY the User's password is set, then this is a power on password and must be entered to boot or enter Setup. In Setup the User will have Administrator rights.
Administrator Password
User Password
→←: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3 Optimized Defaults
F4: Save ESC: Exit
Version 2.00.1201. Copyright (C) 2009 American Megatrends, Inc.

The following modes of security are provided:

SETTING	DESCRIPTION
No password is set	Booting the system as well as entering the Setup is unsecured.
Only Administrator password is set	Booting the system is unsecured. If no valid Administrator password is entered, only limited access to Setup is provided.
Only User password is set	The password is required for booting the system as well as for entering the Setup menu. On every startup, the user will be asked for the password.
Both User and Administrator passwords are set	Booting the system is unsecured. For entering the Setup, a password is required. If the User password is entered here, limited access to the Setup is granted. Entering the Administrator password provides full access to all Setup entries.

Note: The AM5020 provides no factory-set passwords.

4.1 Administrator Password

This function is used to set, change or delete the Administrator password. If there is already a password installed, the system asks for this first. To clear a password, simply enter nothing and acknowledge by pressing Return. To set a password, enter it twice and acknowledge by pressing Return.

Note: The password is case sensitive.

4.2 User Password

This function is used to set, change or delete the User password. If there is already a password installed, the system asks for this first. To clear a password, simply enter nothing and acknowledge by pressing Return. To set a password, enter it twice and acknowledge by pressing Return.

Note: The password is case sensitive.

4.3 Remember the Password

It is highly recommended to keep a record of all passwords in a safe place. Forgotten passwords may lead to being completely locked out of the system. Booting may not be possible, and in worst case the uEFI BIOS Setup program will also not be accessible.

If the system cannot be booted because neither the User password nor the Administrator password are known, refer to Chapter 4.1 in the AM5020 User Guide for information about clearing the uEFI BIOS settings, or contact Kontron for further assistance.

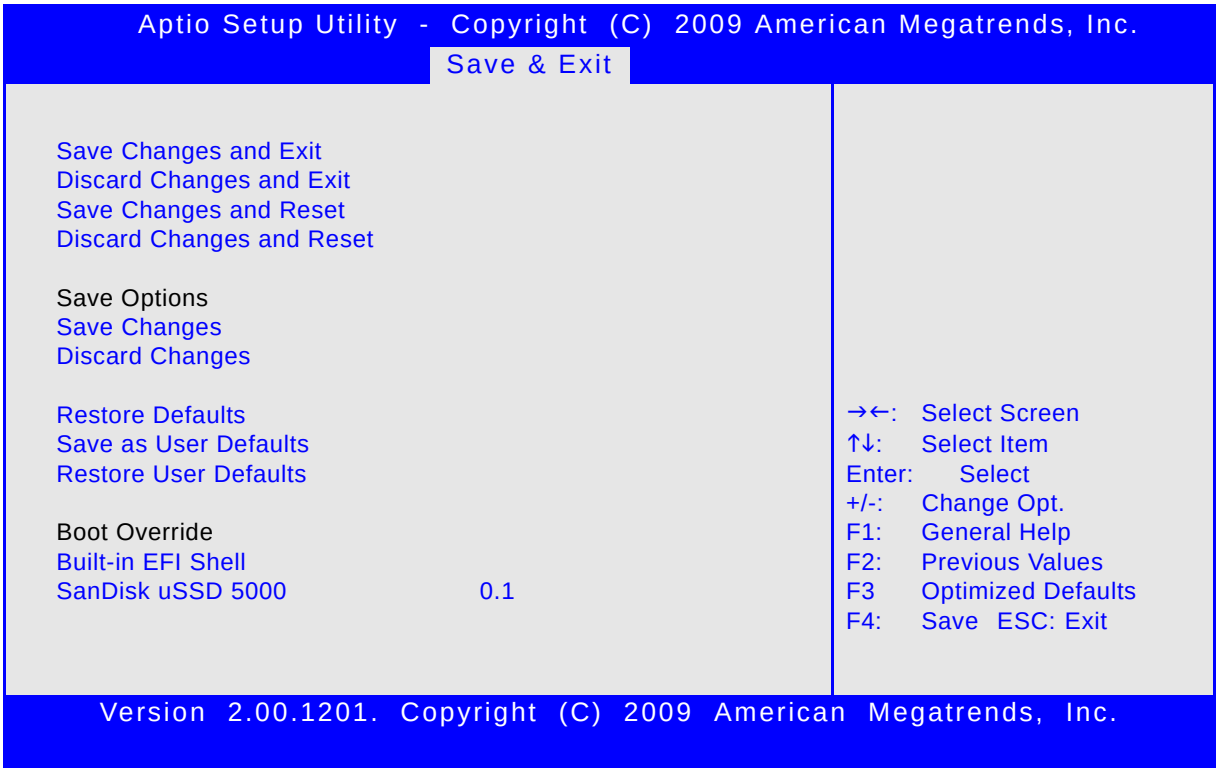
Chapter 5

Save & Exit

This page has been intentionally left blank.

5. Save & Exit

Select the Save & Exit tab to enter the Save & Exit menu screen. This screen provides functions for handling changes made to the uEFI BIOS settings and the exiting of the Setup program.



5.1 Save Changes and Exit

This function is used to save all changes made within the Setup to flash. This function continues the boot process as long as no option was altered that requires a reboot.

Note: The Setup will ask for confirmation prior to executing this command.

5.2 Discard Changes and Exit

This function is used to discard all changes made within the Setup. This function continues the boot process.

Note: The Setup will ask for confirmation prior to executing this command.

5.3 Save Changes and Reset

This function is used to save all changes made within the Setup to flash. This function performs a reboot afterwards.

Note: The Setup will ask for confirmation prior to executing this command.

5.4 Discard Changes and Reset

This function is used to discard all changes made within the Setup. This function performs a reboot afterwards.

Note: The Setup will ask for confirmation prior to executing this command.

5.5 Save Changes (Save Options)

This function is used to save all changes made within the Setup to flash. This function returns to Setup.

Note: The Setup will ask for confirmation prior to executing this command.

5.6 Discard Changes (Save Options)

This function is used to discard all changes made within the Setup. This function returns to Setup.

Note: The Setup will ask for confirmation prior to executing this command.

5.7 Restore Defaults (Save Options)

This function is used to restore all tokens to factory default.

Note: The Setup will ask for confirmation prior to executing this command.

5.8 Save as User Defaults (Save Options)

This function is used to save all current settings as user default. The current setup state can later be restored using Restore User Defaults.

Note: The Setup will ask for confirmation prior to executing this command.

5.9 Restore User Defaults (Save Options)

This function is used to restore all tokens to settings previously stored by Save as User Defaults.

Note: The Setup will ask for confirmation prior to executing this command.

5.10 Boot Override

This group of functions includes a list of tokens, each of them corresponding to one device within the boot order. Select a drive to immediately boot that device regardless of the current boot order. If booting to EFI Shell this way, an exit from the shell returns to Setup.

Chapter 6

The uEFI Shell

This page has been intentionally left blank.

6. The uEFI Shell

The Kontron uEFI BIOS features a built-in and enhanced version of the uEFI Shell. For a detailed description of the available standard shell scripting refer to the EFI Shell User's Guide. For a detailed description of the available standard shell commands, refer to the Shell Command Manual 1.0. Both documents can be downloaded from the EFI and Framework Open Source Community homepage (https://efi-shell.tianocore.org) under the "Documents and Files" section.

Please note that not all shell commands described in the Shell Command Manual 1.0 are provided by the Kontron uEFI BIOS.

6.1 Introduction, Basic Operation

The uEFI Shell forms an entry into the uEFI boot order and is the first boot option by default. It is simply started by putting the uEFI Shell first in boot and running the board as usual.

6.1.1 Shell Startup

If the shell is executed, it displays its signon message followed by a list of detected devices. The output produced by the device mapping table can vary depending on the board's configuration.

```
EFI Shell version 2.00 [4.631]
Current running mode 1.1.2
Device mapping table
  fs0     :Removable HardDisk - Alias hd33b0b0b blk0
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)/HD(Part1,Sig17731773)
  fs1     :Removable BlockDevice - Alias f33b0c0 blk1
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(2, 0)
  blk0    :Removable HardDisk - Alias hd33b0b0b fs0
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)/HD(Part1,Sig17731773)
  blk1    :Removable BlockDevice - Alias f33b0c0 fs1
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(2, 0)
  blk2    :HardDisk - Alias (null)
           Acpi(PNP0A03,0)/Pci(1F|2)/Ata(Primary,Master)/HD(Part1,SigC811D18D)
  blk3    :BlockDevice - Alias (null)
           Acpi(PNP0A03,0)/Pci(1F|2)/Ata(Primary,Master)
  blk4    :Removable BlockDevice - Alias (null)
           Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)

Press the ESC key within 5 seconds to skip startup.nsh, and any other key to
continue.
```

If the ESC key is pressed before the 5-second timeout has elapsed, the shell prompt is shown:

```
Shell>
```

6.2 Kontron-Specific uEFI Shell Commands

The Kontron uEFI implementation provides the following additional commands related to the specific HW features of the Kontron system:

- **kboardconfig**
- **kboardinfo**
- **kboot**
- **kbootnsh**
- **kclearnvram**
- **kclsp**
- **kflash**
- **kipmi**
- **kmkramdisk**
- **kpassword**
- **kpci**
- **kwdt**

The following tables provide information concerning these Kontron-specific commands. The command response values indicated can vary depending on the board’s configuration.

6.2.1 kboardconfig uEFI Shell Command

kboardconfig

FUNCTION:	Configure the non-volatile board settings
SYNTAX:	**kboardconfig** **kboardconfig [-?|<device>|<setting>]** where: ? Show online help <device> Specify device from list <setting> Select configuration type
DESCRIPTION:	The **kboardconfig** command is used to configure non-volatile board settings.
USAGE:	Show all possible configurations COMMAND / RESPONSE: **Shell> kboardconfig** **Control nonvolatile board settings** **Example: kboardconfig** **pxe: Select PXE boot network adapter ([disabled] all front_a front_b amc_0 amc_1)** **StorageOrom: Launch Storage PCI OpROM (disabled [enabled])** **HyperThreading: Enable Hyper Threading technology (disabled [enabled])** **CpuTurbo: Enable CPU turbo mode technology (disabled [enabled]** **PrimaryDisplay: Select primary display device ([auto] igd peg pci)** **SataMode: Determines how SATA controller(s) operate ([ide] ahci raid)** **wr_prot_eeprom: System EEprom write protection ([disabled] enabled)** **wr_prot_sata: Onboard Sata flash write protection ([disabled] enabled)** **wr_prot_spi: EFI spi flash write protection ([disabled] enabled)**
	Show allowed settings e.g. for “PrimaryDisplay”: **Shell> kboardconfig PrimaryDisplay** **PrimaryDisplay: Select primary display device** **PrimaryDisplay == auto** **Allowed options: auto, igd, peg, pci**
SETTINGS:	**pxe:** Select PXE boot network adapter **disabled:** No PXE boot available **[all]:** Try all Ethernet devices round robin for PXE boot **front_a:** Try only front port a for PXE boot **front_b:** Try only front port b for PXE boot **amc_0:** Try only AMC port 0 for PXE boot **amc_1:** Try only AMC port 1 for PXE boot Note: **front_a** corresponds to GbE C and **front_b** corresponds to GbE D on the front panel of the AM5020.

kboardconfig (continued)

SETTINGS	**StorageOrom:** Launch Storage PCI Option ROMs **disabled:** Do not launch storage PCI option ROMs. This includes the onboard RAID option ROM. **[enabled]:** Launch storage option ROMs, if present
	HyperThreading: Enable/Disable Hyper-Threading Technology
	CpuTurbo: Enable/Disable CPU Turbo Boost Technology
	PrimaryDisplay: Select primary display device **[auto]:** Automatically detect primary display device **igd:** Use internal graphics, if enabled **peg:** Try to use video on the PCIe graphics port, if present **pci:** Try to use video on the PCI(e) bus first
	SataMode: Determines how SATA controllers operate **[ide]:** SATA ports operate as two IDE controllers **ahci:** SATA ports operate as one 6-port AHCI controller **raid:** SATA ports form a RAID device
	wr_prot_eeprom: System EEPROM write protection **[disabled]:** Do not write protect the system EEPROM **enabled:** System EEPROM is write-protected after POST
	wr_prot_sata: Onboard SATA flash write protection **[disabled]:** Do not write protect the onboard SATA flash **enabled:** The onboard SATA flash is write-protected after POST. OS needs to be prepared to work with write-protected flash. For further information, refer to the operating system's documentation.
	wr_prot_spi: EFI SPI flash write protection **[disabled]:** Do not write protect the EFI SPI flash **enabled:** The EFI SPI flash is write-protected after POST

6.2.2 kboardinfo uEFI Shell Command

kboardinfo

FUNCTION:	Show board identification data
SYNTAX:	**kboardinfo**
DESCRIPTION:	The **kboardinfo** command shows a summary of board-specific identification data. It is especially useful for support queries because it contains this data in a concentrated form.
USAGE:	Show board identification data COMMAND / RESPONSE: **Shell> kboardinfo** **KOMaOEMF rev.: 3** **Board ID: 0xB300** **Hardware rev.: 0x0** **Logic rev.: 0x9** **Boot flash: Boot flash 0** **In system slot: Yes** **Geographic address: 3** **Material number:** **Hardware index:** **Serial number:** **EFI article name: SK-EFI-B3001** **EFI material number:** **EFI index: 13, standard** **EFI build time: 16:37:05** **EFI build date: 12/30/2010** **NorthBridge rev.: 0x12** **SouthBridge rev.: 0x6** **Microcode: 0x9** **CPU ID: 0x20652** **CPU Branding: Intel(R) Core(TM) i7 CPU L 620 @ 2.00 GHz**

kboardinfo (continued)

KOMaOEMF rev.:	Revision of KOMaOEMF protocol
Board ID:	Kontron board identification value (should be 0xB300 for the AM5020)
Hardware rev.:	Hardware revision of this board
Logic rev.:	Logic revision of this board
Boot flash:	Current boot flash: either “Boot flash 0” or “Boot flash 1”
In system slot:	This setting is not relevant.
Geographic Address:	Geographic address of the MicroTCA back-plane slot the board is currently plugged into
Material number:	Kontron hardware reference number
Hardware index:	Kontron hardware index
Serial number:	This board’s unique serial number
EFI article name:	Kontron uEFI reference name
EFI material number:	Kontron uEFI reference number
EFI index:	Version of this uEFI BIOS
EFI build time:	Build time of this uEFI BIOS
EFI build date:	Build date of this uEFI BIOS
NorthBridge rev.:	Chip revision of the NorthBridge (GMCH die of the Intel® Core™ i7 processor)
SouthBridge rev.:	Chip revision of the SouthBridge (Intel ® QM57)
Microcode:	Currently loaded microcode
CPU ID:	CPUID
CPU Branding:	CPU identification string

6.2.3 kboot uEFI Shell Command

kboot

FUNCTION:	Boot a legacy OS Not to be used for uEFI BootLoaders!
SYNTAX:	**`kboot [-?|-d|-p|-p <path>|-n <name>|-t <type>]`** where: ? Show online help -d Boot default order -p <path> Specify the path to the device to boot from -n <name> Specify the device name to boot from -t <type> Specify the device type to boot from Available types are: floppy harddrive cdrom network usb-floppy usb-harddrive usb-cdrom
DESCRIPTION:	The **kboot** command boots a legacy OS. Boot device can be selected in a very flexible way. If the requested device is not present, boot returns to shell. The **kboot** command cannot boot native uEFI-aware operating systems. But since these are bootable from shell by calling their bootloader, this is not necessary either. If a requested device is present but not bootable, uEFI continues to boot with the next bootable device in the boot order.

kboot (continued)

USAGE:

Show all connected devices:

COMMAND / RESPONSE:

```
fs0:\> kboot
____BBS_TABLE____
00002 network “IBA GE Slot 0100 v1300”
00003 network “IBA GE Slot 0101 v1300”
00004 network “IBA GE Slot 0200 v1300”
00005 network “IBA GE Slot 0201 v1300”
00002 usb-harddrive “SanDisk uSSD 5000 0.1”
Device path: Acpi(PNP0A03,0)/Pci(1A|7)/Usb(1,0)
0001 usb-harddrive “KingstonDataTraveler 2.04.10”
Device path: Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1,0)
```

Boot from device containing the string “Kingston”:

```
fs0:\> kboot -n Kingston
```

Boot from the first device found that is of type floppy:

```
fs0:\> kboot -t floppy
```

6.2.4 kbootnsh uEFI Shell Command

kbootnsh

FUNCTION:	Manage the startup script stored in the flash
SYNTAX:	**`kbootnsh [-b][-?|-g <filename>|-p <filename>|-d]`** where: -b Display output page by page -? Show online help -g <filename> Store the current boot script to disk. If there is no physical disk drive present, the **kmkramdisk** command may be used. -p <filename> Store the shell script pointed to by filename to flash. Note: The shell script cannot be larger then 400 bytes. -d Delete the current startup script from flash.
DESCRIPTION:	The **kbootnsh** command manages the flash stored startup script. If the shell is launched by the boot process, it executes a shell script stored in the flash. If the shell script terminates, the shell executes a **kboot -d** command to continue the boot process. However, the shell script can of course contain any other boot command.
USAGE:	Get current startup script to file named boot.nsh COMMAND / RESPONSE: **`Shell> kbootnsh -g boot.nsh`**
	Store file named boot.nsh to flash: COMMAND / RESPONSE: **`Shell> kbootnsh -p boot.nsh`**
	Delete startup script: COMMAND / RESPONSE: **`Shell> kbootnsh -d`**

6.2.5 kclearnvram uEFI Shell Command

kclearnvram

FUNCTION:	Clear the NVRAM to restore the system's default settings
SYNTAX:	**`kclearnvram`** No parameters required. For safety reasons this command must be confirmed by pressing "c".
DESCRIPTION:	The **kclearnvram** command allows to clear the system NVRAM. Since all EFI settings are stored inside the NVRAM, the default settings are loaded afterwards.

6.2.6 kclsp uEFI Shell Command

kclsp

FUNCTION:	Configure clock spreading
SYNTAX:	**`kclsp [-?|-d|-e]`** where: -? show help -d disable clock spreading -e enable clock spreading
DESCRIPTION:	The **kclsp** command enables or disables clock spreading on the onboard core clock generator. Clock spreading can be used to reduce system EMI.
USAGE:	Get help: COMMAND / RESPONSE: **`Shell> kclsp -?`** **`Kontron Clock Spreading Configuration for ICS9LPRS365`** **`-d disable clock spreading`** **`-e enable clock spreading`** Default setting: disable

6.2.7 kflash uEFI Shell Command

kflash

FUNCTION:	Manage uEFI BIOS update
SYNTAX:	**`kflash [-p|-i|-v|-s|-c|-h|-?] [-f] [-r] [file]`** Operation mode: -p Program flash -i Show information string and check CRC -v Verify flashed image -s Save current ROM image to file -c Clone flash content to second flash -h Show this help -? Show online help file uEFI BIOS binary file Options: -f Force write (option) Expert options: Not recommended for standard use -r Raw image mode (.bin, .rom)
DESCRIPTION:	The **kflash** command is used to program and verify the flash banks holding the uEFI BIOS code. uEFI BIOS binary files must be available from connected mass storage devices, such as USB flash drive or harddisk.
USAGE:	Get help: COMMAND / RESPONSE: **`Shell> kflash -?`**
	Get help: COMMAND / RESPONSE: **`Shell> kflash -h`**
	Program uEFI BIOS into flash bank: COMMAND / RESPONSE: **`Shell> kflash -p BIOS_file.kfl`**
	Copy uEFI BIOS in secondary BIOS bank: COMMAND / RESPONSE: **`Shell> kflash -c`**

6.2.8 kipmi uEFI Shell Command

kipmi

FUNCTION:	Read or configure available MMC parameters
SYNTAX:	**`kipmi [-?|-b|parameters]`** where: -? show online help -b display output page by page parameters fru -- display fru data: [Fru Device ID] ipmb -- ipmb bus settings: ipmb [redundant / ssingle] irq -- get / set KCS IRQ: irq [number] mode -- set ipmi controller mode: mode [bmc / smc] net -- display and change SOL network settings sel -- handle system event log sensor -- shows sensor related information raw -- execute raw ipmi command rawsendmessage -- execute raw SendMessage ipmi cmd info -- show information about the device and firmware
DESCRIPTION:	The **kipmi** command can read event logs or set the MMC IRQ configuration. This shell application can also be used to set up raw command to the MMC.
USAGE:	Display fru data: COMMAND / RESPONSE: **`Shell> kipmi fru 0`**
	Display ipmb bus settings: COMMAND / RESPONSE: **`Shell> kipmi ipmb`**
	Change IRQ configuration: COMMAND / RESPONSE: **`Shell> kipmi irq 10`**
	Show IRQ configuration: COMMAND / RESPONSE: **`Shell> kipmi irq`**

kipmi (continued)

	Set IPMI controller mode: COMMAND / RESPONSE: **`Shell> kipmi mode`**
USAGE:	Set Serial-over-LAN I/O/SOL parameters: COMMAND / RESPONSE: **`Shell> kipmi net 1`**
	Display system event log: COMMAND / RESPONSE: **`Shell> kipmi sel list`**
	Show sensor related information: COMMAND / RESPONSE: **`Shell> kipmi sensor list`**
	Execute raw command. Example: Get self-test results. COMMAND / RESPONSE: **`Shell> kipmi raw 0x06 0x00 0x04`**
	Execute raw SendMessage command: COMMAND / RESPONSE: **`Shell> kipmi rawsendmessage 0x20 0x00 0x06 0x00 0x01`**
SETTINGS:	**`fru [<Fru device ID>]:`** Displays FRU data Options: **`Fru device ID:`** Numeric FRU device ID. The FRU ID 0 is used by default if no FRU ID is entered.
	`ipmb:` Displays IPMB bus settings **`ipmb redundant:`** Switch IPMB bus to redundant mode **`ipmb single:`** Switch IPMB bus to single mode Note: The redundant mode is not available on the AM5020. Please leave this function at single mode.
	`irq <number>:` Display/Set the IRQ number of the KCS interface Options: **`0:`** KCS uses no IRQ **`10:`** KCS uses IRQ 10 **`11:`** KCS uses IRQ 11 The board must be reset for the settings to apply.

kipmi (continued)

SETTINGS:	**mode <mode>:** Display/Set the IPMI controller (MMC) operating mode Options: **bmc:** IPMI controller (MMC) operates in BMC mode (master) **smc:** IPMI controller (MMC) operates in SMC mode (slave) Note: The BMC mode is not available on the AM5020. Please leave this function at SMC mode.
	net: Set IPMI-over-LAN (IOL) / Serial-over-LAN (SOL) parameters
	sel: Display system event log Note: The AM5020 does not have a system event log.
	sensor list|read: Show board sensor data Options: **list:** Display an overview of all available board sensors **read:** Display specific sensor data
	raw [<bytes> <...>]: Execute raw IPMI command Syntax: **raw [NetFn] [LUN] [COMMAND] ...**
	rawsendmessage [<bytes> <...>]: Execute raw SendMessage command Syntax: **rawsendmessage [rsSa] [CHANNEL] [NetFn] [LUN] [COMMAND] ...**
	info: Display IPMI firmware information

6.2.9 kmkramdisk uEFI Shell Command

kmkramdisk

FUNCTION:	Create RAMdisk drives
SYNTAX:	**kmkramdisk [-?|-s <size> <name>]** where: -? show help -s <size> <name> create a RAMdisk of given size in Megabytes with the mount point name <name>
DESCRIPTION:	Creates a RAMdisk of variable size. Can be very useful to perform file operations when no real filesystem is connected to the system. Note: The RAMdisk loses its mount point name after all drives are remapped by the **map -r** command. The RAMdisk will then be enumerated as any other connected drive and gain a mount point name like “fs0”. This is not a bug of the **kmkramdisk** command but a normal function of the uEFI framework.
USAGE:	Create RAMdisk: COMMAND / RESPONSE: **rd:\> kmkramdisk -s 5 myramdisk** **Device mapping table** **myramdisk :BlockDevice - Alias (null)** **VenMsg’(93B5F448-127A-4B29-B306-5BE8AAC4826E)** **Success - Force file system to mount** **rd:\> myramdisk:** **myramdisk:\> echo testfile > testfile** **myramdisk:\> ls** **Directory of: myramdisk:** **05/24/08 04:39a 22 testfile** **1 File(s) 22 bytes** **0 Dir(s)**

6.2.10 kpassword uEFI Shell Command

kpassword

FUNCTION:	Control EFI setup and shell passwords
SYNTAX:	**kpassword [-u|-s]** Call without parameters to get current password status Parameters: -u Install or change user password -s Install or change superuser password Note: Old passwords must be verified if set. Entering an empty password disables the password.
DESCRIPTION:	The **kpassword** command is used to get and set the EFI shell and setup passwords. Both user and superuser (Administrator) passwords can be controlled.
USAGE:	Control EFI setup and shell passwords COMMAND / RESPONSE: **kpassword [-u|-s]** **No password is installed!** **Enter new USER password** **-->** **Retype password** **-->** **Done.**

6.2.11 kpci uEFI Shell Command

kpci

FUNCTION:	Control PCI Express configuration
SYNTAX:	**`kpci [-b|-v|-x|-h|-?][parameter]`** where: -b Display output page by page -v Be more verbose -x Show transferred data in hex -h Show this help ? Show this help Available parameters: pcie -- PCI Express port lane configuration
DESCRIPTION:	The **kpci** command can control the behavior of the AMC PCI Express port and keep this configuration in sync with FRU data offered by the IPMI controller during E-Keying.
USAGE:	Display PCIe port(s) setting: COMMAND / RESPONSE: **`Shell> kpci pcie`**
	Set PCIe port(s) to be two times x4: COMMAND / RESPONSE: **`Shell> kpcie pcie 1x4`** Note: If the PCI Express setting in the uEFI BIOS does not correspond to that in the MMC, a warning message for 5 seconds is displayed. Please use this command to correct this condition.
	Set PCIe port(s) to be eight times x1 COMMAND / RESPONSE: **`Shell> kpcie pcie 4x1`** Note: If the PCI Express setting in the uEFI BIOS does not correspond to that in the MMC, a warning message for 5 seconds is displayed. Please use this command to correct this condition.

6.2.12 kwdt uEFI Shell Command

kwdt

FUNCTION:	Configure the Kontron onboard Watchdog
SYNTAX:	**kwdt [-?|-t <timeindex>]** where: -? Show help -t <timeindex> Configure the Watchdog with the time related to timeindex and activate it with reset routing Call kwdt -h to obtain a list of time index values and related times
DESCRIPTION:	The **kwdt** command allows to enable the Kontron onboard Watchdog with reset target before OS boot. This can be used to detect if the OS fails to boot and react by reset. The OS Watchdog driver is required for this functionality to operate.
USAGE:	Get help: COMMAND / RESPONSE: **Shell> kwdt -?** **-t [time] - set Timer** **value 0 = 125ms** **value 1 = 250ms** **value 2 = 500ms** **value 3 = 1s** **value 4 = 2s** **value 5 = 4s** **value 6 = 8s** **value 7 = 16s** **value 8 = 32s** **value 9 = 64s** **value 10 = 128s** **value 11 = 256s** **value 12 = 512s** **value 13 = 1024s** **value 14 = 2048s** **value 15 = 4096s**

kwdt (continued)

Set Watchdog to 16 seconds and activate it

COMMAND / RESPONSE (none):

```
Shell> kwdt -t 7
```

Note: Because there is no application which triggers the Watchdog, the system will be reset after 16 seconds in this case. This command should be invoked from a script, followed by an operating system boot, and the OS then has to start triggering the Watchdog.

Display Watchdog configuration:

COMMAND / RESPONSE:

```
Shell> kwdt
Kontron Board Watchdog Configuration:
Watchdog Configuration Register (0x28C):   0x00
```

6.3 uEFI Shell Scripting

6.3.1 Startup Scripting

If the ESC key is not pressed and the timeout is run out, the uEFI Shell tries to execute some startup scripts automatically. It searches for scripts and executes them in the following order:

1. Kontron flash-stored startup script
2. If there is no Kontron flash-stored startup script present, the uEFI-specified **`startup.nsh`** script is used. This script must be located on any of the attached FAT formatted disk drives under **`\efi\boot\startup.nsh.`**
3. If none of the startup scripts is present or the startup script terminates, the default boot order is continued.

6.3.2 Create a Startup Script

Startup scripts can be created using the uEFI Shell built-in editor **edit** or under any OS with a plain text editor of your choice. To create a startup shell script, simply save the script on any FAT-formatted drive attached to the system under the file name **`\efi\boot\startup.nsh.`** To copy the startup script to the flash use the **kbootnsh** uEFI Shell command.

In case there is no mass storage device attached, the startup script can be generated in a RAM disk and stored in the flash bank.

6.3.3 Examples of Startup Scripts

6.3.3.1 Automatic Booting from USB Flash Drive

Automatic booting is made from a USB flash drive, if present, otherwise the boot is made from the harddrive.

```
kboot -t usb-harddrive
kboot -t harddrive
```

If neither a USB flash drive nor a harddrive is present, the boot order is continued.

6.3.3.2 Switch On Clock Spreading Prior to Booting from Harddrive

```
kclsp -e
kboot -t harddrive
```

If no harddrive is present, the default order is continued.

6.3.3.3 Execute Shell Script on Other Harddrive

This example executes the shell script named **`bootme.nsh`** located in the root of the first detected disc drive (**`fs0`**).

```
fs0:
bootme.nsh
```

6.3.3.4 Enable Watchdog and Control PXE Boot

The uEFI Shell provides environment variables used to control the execution flow.

The following sample start-up script shows two uEFI Shell environment variables, **wdt_enable** and **pxe_first**, used to control the boot process and the Watchdog.

```
echo -off
echo “Executing sample startup.nsh...”
if %wdt_enable% == “on” then
   kwdt -t 15
   echo “Watchdog enabled”
endif
if %pxe_first% == “on” then
   echo “forced booting from network”
   kboot -t network
endif
```

To create uEFI Shell environment variables, use the **set** uEFI Shell command as shown below:

```
Shell> set wdt_enable on
Shell> set pxe_first on
Shell> set
   pxe_first : on
   wdt_enable : on
Shell> reset
```

6.3.3.5 Handling the Startup Script in the Flash Bank

In case there is no mass storage device attached, the startup script can be generated in a RAM disk and stored in the flash bank using the following instructions:

4. Press <ESC> during power-up to log into the uEFI Shell.
5. Create a RAM disk and set the proper working directory as shown below:

```
Shell> kmkramdisk -s 3 myramdisk
Shell> myramdisk:
```

6. Enter the sample start-up script mentioned above in this section using the **edit** uEFI Shell command.

```
myramdisk:\> edit boot.nsh
```

7. Save the start-up script to the uEFI flash bank using the **kbootnsh** uEFI Shell command.

```
myramdisk:\> kbootnsh -p boot.nsh
```

8. Reset the board to execute the newly installed script using the **reset** uEFI Shell command.

```
myramdisk:\> reset
```

9. If a script is already installed, it can be edited using the following **kbootnsh** uEFI Shell commands.

```
myramdisk:\> kbootnsh -g boot.nsh
myramdisk:\> edit boot.nsh
```

Chapter 7

Updating the uEFI BIOS

This page has been intentionally left blank.

7. Updating the uEFI BIOS

BIOS updates are typically delivered as an update CD ISO image. This ISO image needs just to be burned to a CD and booted. Follow the menu for updating the uEFI BIOS. For further information refer to the update CD documentation.

7.1 BIOS Redundancy Strategy

The AM5020 has two sets of uEFI flash banks to form a redundancy strategy. The basic idea behind that is to always have at least one working uEFI Flash bank available regardless if there have been any flashing errors or not.

7.2 Updating Strategy

To always maintain at least one uEFI flash correct, the update CD uses the following update procedure:

1. Switch to the second flash bank.
   Since the update CD always changes the flash bank prior to doing any updates, the uEFI BIOS that was used to actually boot the board and is therefore known to be good is preserved for backup.
2. Update the second flash bank.
   This flash is now selected as active boot flash.

The update CD will not allow to flash both banks at a time. Flashing both banks would destroy the backup version and therefore break the redundancy.

If you want to have the same BIOS version on both flash banks, then simply run the update CD twice.

7.3 uEFI BIOS Recovery

In case the uEFI BIOS update has failed due to power loss, the MMC firmware will automatically fall back to a redundant uEFI BIOS flash bank. A new attempt to perform the uEFI BIOS update can be started once the power is back on.

If the updated uEFI BIOS is not suitable for the system, the selection of the redundant flash bank must be done manually. An example of a reason for unsuitability is the use of an older uEFI BIOS version with missing features.

7.4 Determining the Active Flash

Sometimes it may be necessary to check which flash is active. In addition to the **`Get Control State`** IPMI OEM command, the **kboardinfo** uEFI Shell command can also be used to determine the currently selected flash bank. For further information, refer to Chapter 6.2.2, kboardinfo uEFI Shell Command.

7.5 Manual Flash Selection

Usually the active flash is selected by the MMC controller. The manual selection of the redundant flash bank can be done either via the **Set Control State** IPMI OEM command or using the DIP Switch SW4, switch 2. The **Set Control State** IPMI OEM command can be issued without having payload power applied.

The flash bank can be switched via one of the following IPMI OEM commands:

To select the first flash bank, use the following **Set Control State** command:

```
impitool raw 0x3E 0x20 0x00 0x00
```

To select the second flash bank, use the following **Set Control State** command:

```
impitool raw 0x3E 0x20 0x00 0x01
```

To determine the currently selected flash bank, use the following **Get Control State** command:

```
impitool raw 0x3E 0x21 0x00
```

For further information regarding the **Set Control State** Commands and the **Get Control State** Command, refer to the IPMI manual, Chapter 3, OEM Commands and Command Extensions.

7.6 Flash Selection by DIP Switch

In some cases it may be necessary to force the board to boot from the other flash bank without using the management software. In this case, the onboard DIP switch SW4, switch 2, is used to toggle the active flash bank. Please note that this switch does not “select” a flash bank. It only toggles the currently selected flash bank.